CLAIMS

What is claimed is:

1. A method, comprising:

dynamically invoking a speculative thread from a main thread in a processor; and

executing instructions comprising the speculative thread.

2. The method of claim 1, further comprising attempting to bind a hardware thread context for the speculative thread.

3. The method of claim 2, further comprising determining whether the attempt to bind the hardware thread context for the speculative thread was a success or a failure.

4. The method of claim 3, further comprising raising an exception and branching to a recovery handler when it is determined that the attempt to bind a hardware thread context for the speculative thread was a success.

5. The method of claim 3, further comprising treating the attempt to bind a hardware thread context for the speculative thread as a no operation instruction when it is determined that the attempt to bind a hardware thread context for the speculative thread was a failure.

6. The method of claim 1, further comprising transferring live-in values for pre-computation slices to hardware thread context register files.

7. The method of claim 6, further comprising allocating a portion of memory for the speculative thread and dedicating the allocated a portion of on-chip memory buffer space as an intermediate buffer to pass live-in values from the main thread to the speculative thread.

8. The method of claim 6, further comprising:

identifying at least one load of interest;

constructing pre-computation slices for each load of interest, wherein the pre-computation slice comprises a speculative thread that pre-computes an address accessed by a load of interest and pre-fetching for the load of interest using the pre-computed address; and establishing triggers to invoke the speculative thread.

9. A method in a processor, comprising:

dynamically invoking from a first speculative thread a second speculative thread, wherein the first speculative thread is dynamically invoked from a main thread; and

executing instructions comprising the first and second speculative threads.

10. The method of claim 9, further comprising attempting to bind a hardware thread context for the speculative thread.

11. The method of claim 9, further comprising detecting a trigger to invoke the second speculative thread, storing second speculative thread live-in values to a buffer, and executing instructions in the second speculative thread.

12. The method of claim 10 wherein each speculative thread includes a pre-computation slice, the method further comprising:

allocating an entry in a queue for a copy of at least one pre-computation slice when it is determined that a hardware context is unavailable for the copy of the pre-computation slice; and

placing the pre-computation slice in the queue until a hardware context is available.

13. A processor, comprising:

a first hardware context to store a main software thread;

a second hardware context to store a speculative software thread;

logic coupled between the first and second hardware contexts and a machine-readable medium having machine-readable instructions stored thereon to instruct a processor to bind the speculative software thread to the second hardware context and to transfer live-in values from main software thread to the speculative software thread; and

14. The processor of claim 13 wherein the logic to copy live-in values from first hardware context to the second hardware context includes flash-copy hardware or portion of memory buffer space.

15. A processor, comprising:

a first hardware context to store a main thread;

a second hardware context and a third hardware context to bind to a first speculative thread and a second speculative thread, respectively, the main thread to dynamically invoke the first speculative thread and the first speculative thread to dynamically invoke the second speculative thread; and logic coupled between the first, second, and third hardware contexts, and a processor-readable medium having processor-readable instructions stored thereon to instruct the processor, to bind the first and second speculative threads to the second and third hardware contexts, respectively, and to transfer live-in values from main thread to the first speculative thread and from the first speculative thread to the second speculative thread, wherein the processor-readable medium includes at least one instruction to instruct the processor to trigger the invocation of the first and second speculative threads.

16. The processor of claim 15, further comprising a pending slice queue having entries to allocate a copy of at least one pre-computation slice when it is determined that a hardware context is unavailable for the copy of the pre-computation slice and to hold the pre-computation slice until a hardware context is available.

17. The processor of claim 15, further comprising an outstanding pre-computation slice counter to track for a set of loads of interest the number speculative threads that have been spawned relative to the number of instances of any load of interest that have not yet been retired by the main thread and to decrement en the main thread retires the corresponding load of interest.

18. The processor of claim 15, further comprising an outstanding pre-computation slice counter to track for a set of loads of interest the number speculative threads that have been spawned relative to the number of instances of any load of interest that have not yet been retired by the main thread and to decrement en the main thread retires the corresponding load of interest, to increment the counter when the corresponding pre-computation slice is spawned, and to force any speculative thread for which the entry in the counter is in a predetermined state to wait in the queue until the entry in the counter becomes a second predetermined state.

19. A machine-readable medium having machine-readable instructions stored thereon to instruct a processor to:

dynamically invoke a speculative thread from a main thread; and

execute instructions comprising the speculative thread.

20. The machine-readable medium of claim 19 wherein the instructions are further to instruct the processor to:

allocate an entry in a queue for a copy of at least one pre-computation slice when it is determined that a hardware context is unavailable for the copy of the pre-computation slice; and

place the pre-computation slice in the queue until a hardware context is available.

21. The machine-readable medium of claim 19 wherein the instructions are further to instruct the processor to:

bind a hardware context for the speculative thread;

transfer live-in values from the main thread to the speculative thread; and

load live-in values in the hardware context for the speculative thread.

22. The machine-readable medium of claim 19 wherein the instructions are further to instruct the processor to:

identify a set of loads of interest;

construct pre-computation slices for each delinquent load of interest, wherein the pre-computation slice comprises a speculative thread that pre-computes an address accessed by a load of interest; and establish triggers to invoke the speculative thread.

23. The machine-readable medium of claim 22 wherein the instructions are further to instruct the processor to:

allocate an entry in a queue for a copy of at least one pre-computation slice when it is determined that a hardware context is unavailable for the copy of the pre-computation slice; and

place the pre-computation slice in the queue until a hardware context is available.

24. The machine-readable medium of claim 23 wherein the instructions are further to instruct the processor to:

track for a subset of the set of loads of interest the number speculative threads that have been spawned relative to the number of instances of any load of interest that have not yet been retired by the main thread; and

decrement a counter when the main thread retires the corresponding load of interest.

25. The machine-readable medium of claim 23 wherein the instructions are further to instruct the processor to:

track for a subset of the set of loads of interest the number speculative threads that have been spawned relative to the number of instances of any load of interest that have not yet been retired by the main thread; and

increment a counter when the corresponding pre-computation slice is spawned; and

forcing any speculative thread for which the entry in the counter is in a predetermined state to wait in the queue until the entry in the counter becomes a second predetermined state.

26. A machine-readable medium having machine-readable instructions stored thereon to instruct a processor to:

dynamically invoke from a first speculative thread a second speculative thread, the first speculative thread to be dynamically invoked from a main thread; and

execute instructions comprising the first and second speculative threads invoke a speculative thread from a main thread.

27. The machine-readable medium of claim 26 wherein the instructions are further to instruct the processor to:

bind a first and second hardware contexts for the first and second speculative threads, respectively; and

transfer live-in values from the main thread to the first hardware context and from the first hardware context to the second hardware context.

28. The machine-readable medium of claim 27 wherein the instructions are further to instruct the processor to:

identify a set of loads of interest;

construct pre-computation slices for each delinquent load of interest, wherein the pre-computation slice comprises a speculative thread that pre-computes an address accessed by a load of interest; and

establish triggers to invoke the speculative thread.

29. The machine-readable medium of claim 28 wherein the instructions are further to instruct the processor to:

allocate an entry in a queue for a copy of at least one pre-computation slice when it is determined that a hardware context is unavailable for the copy of the pre-computation slice; and

place the pre-computation slice in the queue until a hardware context is available.

30. The machine-readable medium of claim 28 wherein the instructions are further to instruct the processor to:

track for a subset of the set of loads of interest the number speculative threads that have been spawned relative to the number of instances of any load of interest that have not yet been retired by the main thread; and

decrement a counter when the main thread retires the corresponding load of interest.

31. The machine-readable medium of claim 28 wherein the instructions are further to instruct the processor to:

track for a subset of the set of loads of interest the number speculative threads that have been spawned relative to the number of instances of any load of interest that have not yet been retired by the main thread; and

increment a counter when the corresponding pre-computation slice is spawned; and

forcing any speculative thread for which the entry in the counter is in a predetermined state to wait in the queue until the entry in the counter becomes a second predetermined state.